CDPOP

USER MANUAL

2011
Version: 1.0
Last Updated: 2011.06.16

Authors:

E. L. Landguth[1], B. K. Hand[1], J. M. Glassy[1,2], S. A. Cushman[3], M. Jacobi[1], and T. J. Julian[1]

1 - University of Montana, Division of Biological Sciences, Missoula, MT, 59812, USA.

2 - Lupine Logic Inc, Missoula, MT, 59802, USA.

3 - U.S. Forest Service, Rocky Mountain Research Station, 2500 S. Pine Knoll Dr., Flagstaff, AZ 86001, USA

Table of Contents

1 Introduction....3
1.1 Changes from CDPOP v0.7....3
1.2 What can CDPOP do....3
1.3 How does CDPOP work....3
2 Getting started....5
2.1 Dependencies....5
2.1.1 Baseline requirements....5
2.1.2 Python on non-windows platforms....6
2.1.3 Python on windows....6
2.1.4 Obtaining NumPy and SciPy....6
2.2 Installation....6
2.2.1 Installing Python, NumPy, and SciPy....6
2.2.2 Installing CDPOP....7
2.2.3 Description of CDPOP files....7
2.3 Example run....7
2.3.1 Command line run....7
2.3.2 GUI run....8
3 Input....10
3.1 Input Files....10
3.2 Model Parameters....11
3.3 Mating Parameters....11
3.4 Dispersal Parameters....12
3.5 Offspring Parameters....13
3.6 Genetic Parameters....13
3.7 CDEVOLVE....15
3.8 CDINFECT....17
3.9 CDCLIMATE....17
4 Output....18
4.1 Individual-based Output Files....18
4.2 Population-based Output Files....18
5 General issues....20
5.1 How to obtain CDPOP....20
5.2 Debugging and troubleshooting....20
5.3 How to cite CDPOP....20
6 References....21
7 Acknowledgements....21

1 Introduction

The goal of this user manual is to explain the technical aspects of the current release of the CDPOP program. CDPOP v1.0 is a major extension of the CDPOP program (Landguth and Cushman 2010). CDPOP is an individual-based program that simulates the influences of landscape structure on emergence of spatial patterns in population genetic data as functions of individual-based movement, breeding, and dispersal.

1.1 Changes from CDPOP v0.7

There are major innovations in v1.0 which were not included in the previously published v0.70. We list below the new functionalities of CDPOP v1.0:

- **Natural selection** is implemented through differential offspring viability as functions of fitness landscapes.
- Gene flow and natural selection can now be simulated in **dynamic landscapes.**
- A **graphical user interface** provides a user friendly platform that enables users to explore, analyze, and model the effects of life-history and differential models of complex landscapes on the genetic structure of populations.
- Sex-specific dispersal.
- Changes of some internal software components have allowed an overall speed increase and to improve program stability.
- Additional movement function option: negative exponential movement.
- Inclusion of both a mating landscape and a dispersal landscape.
- Vertical transmission of an infection.
- mtDNA option.
- Output genotype option in a general genetic format.

1.2 What can CDPOP do

CDPOP's realistic representation of the spatial environment and population genetic processes provide a powerful framework to investigate the impact of ecological factors on the genetic structure of populations. This approach has already advanced knowledge of the patterns of genetic variation in spatially-explicit contexts (Landguth et al 2010a; Landguth et al 2010b; Cushman and Landguth 2010). Example simulations have included:

- Quantifying the time to detect barriers.
- Correlating migration rates and landscape resistance barriers.
- Testing for the effects of population sample size and number of markers.

1.3 How does CDPOP work

CDPOP v1.0 models genetic exchange for a given resistance surface and *n – (x, y)* located individuals as functions of individual-based movement through mating and dispersal, vital dynamics, and mutation. A user must specify the input parameters through a graphical user interface or input script file. As the model simulates stochastic processes, most applications will quantify mean and variability of genetic structure across many runs. Thus, a Monte Carlo option is provided for the user to choose the number of runs to simulate given a single set of input parameters. In addition, a user may also frequently wish to launch several runs with different parameter values simultaneously (i.e., sensitivity analysis). This functionality is provided through batch capability.

The simulation program assumes constant population density over time. Individuals are assumed to occupy a fixed grid on the landscape that is user defined by the *n – (x,y)* located individuals. The genotype of each locus for each individual can be initialized by randomly choosing from a file containing allele frequencies for each locus, or by reading in a file containing the initial multi-locus genotypes of all the individuals. The initial age structure of the population is specified by an input file specifying initial age frequency. The sex of each initial individual is randomly assigned.

There are five movement functions that define how individuals choose a mate and disperse on the landscape as a function of cost distance: linear, inverse square, negative exponential, nearest-neighbor, and random mixing. With the nearest-neighbor movement function, an individual moves to the available grid location nearest its initial location. Random mixing moves an individual to a grid location that is randomly chosen from the *n* grids in the population. In linear, inverse-square, and negative exponential movement functions, individuals move a distance from their initial location based on a draw from a probability distribution inversely proportional to a linear, inverse-square, or negative exponential function. The user specifies the maximum dispersal distance (in cost units) an individual can travel on the landscape. The probability is one at no distance from the original location and goes to zero at the maximum dispersal distance.

Reproduction is defined by the user as either hermaphroditic or heterosexual. With hermaphroditic mating, there are no distinct sexes, but individuals mate with other individuals according to the movement function choice, exchanging genes in Mendelian reproduction. In heterosexual reproduction, mated pairs are one male to possibly many or no females, and the end of the mating process occurs when all females have mated. Each mated pair can have a number of offspring that is a bounded random draw based on a uniform probability distribution, a Poisson draw with specified mean, or a constant number. Mendelian inheritance with *k*-allele mutation (rate chosen by the user) is used to generate the offspring's genotype and the sex assignment is random.

Dispersal of offspring occurs from the mother's *(x,y)* location according to the selected movement function and the sex of the individual. The vital rates (birth and death) define whether or not the population will have emigrants or immigrants.

Simulating natural selection. Past versions of CDPOP modeled three sources of genetic variation: gene flow, genetic drift, and mutation. These versions assumed that different genotypes have an equal probability of surviving and passing on their alleles to future generations and thus, natural selection was not operating. CDPOP V 1.0 implements natural selection analogously to the adaptive or fitness landscape of allele frequencies (Wright 1932). This new functionality enables extension of landscape genetic analyses to explicitly investigate the links between gene flow and selection in complex landscapes at an individual's level. The user specifies fitness landscape surfaces for each genotype of a single diallelic locus that is under selection. For example, three relative fitness surfaces must be specified for the three genotypes, AA, Aa, and aa, from the two alleles, A and a. Selection is implemented through differential survival of dispersing individuals as a function of the relative fitness at the location on that surface where the dispersing individual settles. CDPOP v1.0 reads and extracts genotype and location specific fitness values for each $n - (x, y)$ individual in the pre-processing step. The program will continue all other processes the same as CDPOP, with an additional step implement selection during the dispersal process.

Simulating dynamic landscapes. The potential impacts of climate change on the connectivity of populations have become an area of concern among scientists and land managers. Current needs include quantitative and spatially-explicit predictions of current and potential future patterns of fragmentation under a range of climate change scenarios (Opdam & Wascher 2003). To address this need, CDPOP v1.0 allows users to input a new landscape surface at a given generation time through new cost distance matrices for both mating and dispersal.

The program is written in Python 2.6 and provided with installation instructions for most platforms, along with sample input files. CDPOP v1.0 is built on a driver-module, plug-in, docking architecture that allows for ease of future modular development. CDPOP v1.0 has been debugged as carefully as possible by testing all combinations of simulation options. Information for users, including user manual, FAQ, publications, ongoing research, developer involvement, and downloads can be found at http://cel.dbs.umt.edu/software/CDPOP/.

2 Getting started

2.1 Dependencies

2.1.1 Baseline Requirements

CDPOP requires the Python2.6.x interpreter, NumPy package, and SciPy

package. Remember that Python modules usually require particular Python interpreters, so be sure the version ID for any external Python module or package (e.g. NumPy or others) matches the version of your Python interpreter (normally v2.6.x).

2.1.2 Python on Non-Windows Platforms

Some common computer platforms come with Python installed. These include MAC OS X and most Linux distributions. To determine which Python a MAC or Linux workstation has installed, start a terminal console and enter "python." You'll see the version number on the top line (enter Control-D to exit). Replacing an older Python interpreter (pre v2.4) with a newer one (v.2.6.x) on a Linux or MAC OS X machine can be tricky, so ask a System Administrator for help if you're not sure which packages depend on the current Python installed.

2.1.3 Python on Windows

Windows (7, XP, 2000, Server) does not come with Python installed, so follow the instructions below to obtain and install Python on a computer running the Windows operating system. Get a windows installation of the base Python installation (current v.2.6.x) at: http://www.python.org/download/releases/.

2.1.4 Obtaining NumPy and SciPy

We recommend using the superpack Windows installer available from the SourceForge website: http://sourceforge.net/project/. Note that more complete information for NumPy is available at www.scipy.org, where the SciPy module is also presented. Another source is http://www.enthought.com/products/epd.php for a free academic and educational usage in a single downloadable installer that has everything and then some (Numpy, Scipy, Matplotlib, and 70+ modules for python).

2.2 Installation

2.2.1 Install Python, NumPy, and SciPy

Make sure that Python and NumPy are installed, and available to you. You can test this by typing "python" at a command window. If python is available you'll get the python prompt ">>>". If it is not a recognized command, it means either that python is installed but is not in your command shell's paths, or that python is not installed. In the first case ask an administrator to add it to your command paths. If your shell locates and loads python, type, "import numpy". Similarly, type, "import scipy". If python does not complain that there are no such modules, all is well.

The following instructions assume Python, NumPy, and SciPy are not yet available on your computer; if they are, skip to section 2.2.2.

* First run the Python executable installer you've chosen (either from www.python.org or ActiveState, accepting defaults for the installation directory. On Windows this will typically place the executables and libraries in c:/Python2.6/bin and the "site-packages" package tree for

user installed Python modules in c:/Python2.6/lib/site-packages. If you are installing it on a network on which you do not have administrative privileges, you may need to ask a system administrator to install python and the NumPy and SciPy packages in their default locations.

* Next install NumPy and SciPy using the supplied executable (superpack) installer or visiting http://www.scipy.org/Download. This will install NumPy and SciPy in your Python ./site-packages directory.

2.2.2 Install CDPOP

Next, install the CDPOP software itself by unpacking the zip archive supplied. Navigate to the directory on your PC where you wish to install CDPOP, and unpack the supplied zip archive file using a free archive tool like 7Zip (7z.exe), Pkunzip, Unzip, or an equivalent. Seven-Zip (7Z.exe) is highly recommended since it can handle all common formats on Windows, MAC OS X and Linux. On Windows, it is best to setup a project specific modeling subdirectory to perform your simulations outside of any folder that has spaces in its name (like "My Documents"). At this point you should be able to execute the supplied test inputs.

2.2.3 Description of CDPOP files

15 files will be installed in your directory. Here is a description of each:

- README.txt – a quick how to run CDPOP instructions
- EDcdmatrix16.csv - example Euclidean distance cost distance matrix file
- xyED16.csv – example n-(x,y) file for individuals
- CDPOP.py - Python driver code and run file
- CDPOP_Disperse.py - Python library for the dispersal functions
- CDPOP_GetMetrics.py - Python library for the metric functions
- CDPOP_Mate.py - Python library of the mating functions
- CDPOP_Modules.py - Python library with general functions
- CDPOP_Offspring.py - Python library for the offspring functions
- CDPOP_PostProcess.py - Python post-processing library
- CDPOP_PreProcess.py - Python pre-processing library
- agedistribution.csv – example age distribution file
- allelefrequency.csv – example allele frequency distribution file
- fitvals0.txt – example fitness landscape for natural selection
- inputvariables16pnts.csv – run parameters corresponding to the example files
- CDPOP_user_manual.pdf – this file

2.3 Example run

2.3.1 Command line run

The example run is for 16-points representing individuals with a cost distance matrix calculated with Euclidean distance. To run the following example, follow these steps:

1. Double check that the 15 files provided in the archive are in the same directory.

2. The included file inputvaribles.csv specifies the parameters that can be changed and used in a sample CDPOP run. Open inputvaribles.csv in your editor of choice. It is in a comma delimited format (each parameter is separated with a comma) and this format can not be changed. A spreadsheet program like Microsoft Excel, allows for easy editing of the tabular values.

3. There will be 3 lines of information in inputvariables.csv: a header line and 2 lines of information corresponding to 2 separate CDPOP runs (batch process). Section 3 contains a breakdown for each column header and the parameters that can be changed.

4. Start the program with a graphical interface (see section 2.3.2) or at the command line as follows: If you use python from the command line, then open a terminal window and change your shell directory to the CDPOP home directory.

5. Run the program: There are a number of ways to run this program. If you are using a command shell you can run the program by typing "python CDPOP.py inputvariables.csv". Note that the program CDPOP is called followed by a system argument, which is the name of the input variable file that you specify with extensions.

6. Check for successful model run completion: The program will provide step-by-step output in the Shell window. Once completed, a simulation time will be printed out and folders batchrun0mcrun0, batchrun1mcrun0, and batchrun1mcrun1 will be created in your CDPOP home directory to store output from the separate batch and/or Monte-Carlo runs. Each of these folders will have a unique date/time stamp preceding 'batchrun0mcrun0' in case you want to run multiple CDPOP runs in this same directory. The program will also provide a log file with program steps and timing events in your CDPOP home directory (cdpop.log).

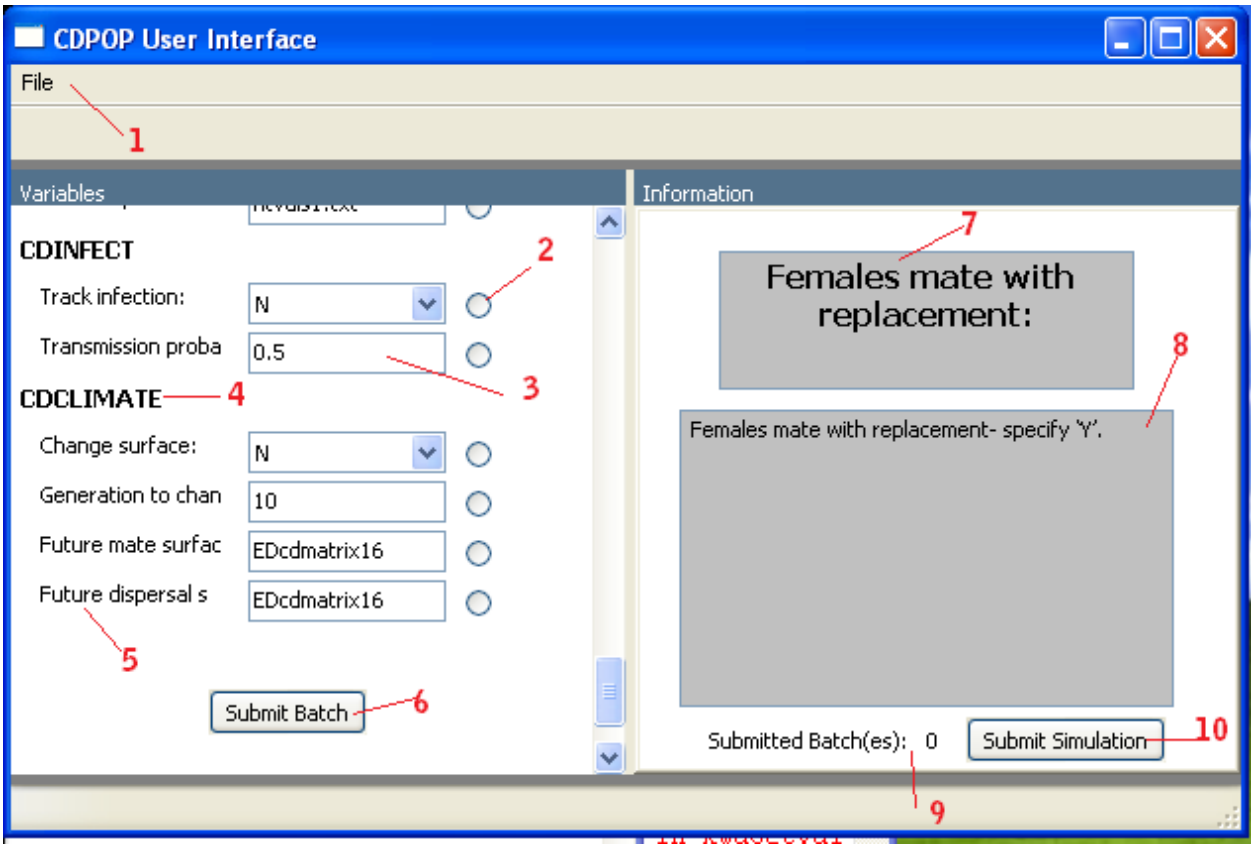
2.3.2 GUI Run

The following are instructions for a simulation run with an optional graphical user interface (GUI).

1) This GUI has a dependency on the python library, WXpython. Go to http://wxpython.org/download.php and download/install your OS's version of WX python.

2) Download and unpack the cdpop_gui.zip: This is a separate download file located under the CDPOP downloads located online. Put all files in the same home directory where you installed CDPOP.

3) Start the interface
   a. Navigate to CDPOP home directory.
   b. Double click cdpopi.py



4) Guide the interface
   a. Clicking 'File' brings down a drop down menu with 'About', and 'Exit'.
      - About – This presents a pop up dialogue of information about the version and developers of CDPOP.
      - Exit – Quits the program. Work is not saved. To prevent lost work, click 'Submit Simulation' before exiting the program.
   b. Radio buttons are listed next to each variable. Only one radio button can be selected at a time. When selected, the 'Information' panel on the right half of the interface displays related information to the associated variable.
   c. These are input fields. There are text input fields and drop down input fields. The default values for each variable are automatically entered.
   d. This is a category header. Related variables are grouped under common headers.

e. Variable name. Identifies what the input field is associated with.
f. Submits a batch of input. A simulation may have 1-N number of batches.
g. Title of selected variable.
h. Detailed description of selected variable.
i. Shows number of currently submitted batches.
j. Submits the simulation with the number of batches shown.

5) Use the interface
a. Enter in your parameters (defaults given are for the example input files).
b. Click 'Submit Batch' in the left panel at bottom.
c. Change parameters if more batches are desired, continuing to click 'Submit Batch'.
d. After satisfied with batch parameter options, click 'Submit Simulation' in the right panel.
e. A successful (or unsuccessful) CDPOP simulation box will notify you and output will be stored in your home directory.

3 Input

3.1 Input files

The following are the general input parameters and files used in CDPOP. See examples provided for formatting. The file header listed is for the first row in the inputvariables.csv describing each file or parameter.

File Header	Example	Description
XY Filename	xyED16	The *n-(x,y)* grid location values. This is a comma delimited file with 5 column headings: (Subpopulation)- a unique identifier for each individual corresponding to a unique subpopulation, (XCOORD)-x-coordinate location, (YCOORD)-y-coordinate location (YCOORD), (ID)-a string label identifier, and (sex)-an initial sex assignment (use 0/1 or F/M). See xyED16.csv for an example xyfilename. The column order is necessary with a header file.
Age Structure Filename	N	The distribution that is used to initialize each individuals age. If 'N' is entered, then this file is not used. If a filename is entered, then read in the file (for example agedistribution would be entered for the example provided). See the agedistribution.csv for formatting this file and note it must be comma delimited. The file includes a row of age classes, number of individuals observed in that age class, and then the proportion of individuals in

		that age class.
Mate CD Matrix Filename	ED-cdmatrix16	A *[nxn]* cost distance matrix for mating movement, where *n* is the number of grid values (or individuals) on the landscape. This is a comma delimited file. See the example edcdmatrix16.csv for formatting this file. Also note that this file can be calculated from any program you choose (e.g., PATHMATRIX, CIRCUITSCAPE, UNICOR, COSTDISTANCE, etc.).
Dispersal CD Matrix Filename	ED-cdmatrix16	A *[nxn]* cost distance matrix for dispersal movement, where *n* is the number of grid values (or individuals) on the landscape. This is a comma delimited file. See the example edcdmatrix16.csv for formatting this file. Also note that this file can be calculated from any program you choose (e.g., PATHMATRIX, CIRCUITSCAPE, UNICOR, COSTDISTANCE, etc.). It can be the same file as the Mate CD Matrix.

3.2 Model parameters

The following lists the general model parameters used for CDPOP.

File Header	Example	Description
Mcruns	1	The repeated number of simulations to be conducted for the Monte Carlo method (i.e., the number of replicates for 1 batch of parameters).
Looptime	5	Simulation run time [generation]. File output indexed from 0 – (looptime-1). For example grid0.csv, grid1.csv, grid2.csv, grid3.csv, and grid4.csv would be outputted for a looptime of 5.
Nthfile Choice	Sequence	The choice of a specified simulation run time [generation] to write to file and to calculate genetic distance matrices. If 'List' is entered, then read nthfile_list values below. If 'Sequence' is entered, then read nthfile_seq value below.
Nthfile List	0|3|4	The specified simulation run time to write to file and to calculate genetic distance matrices. These values are used if nthfile_choice = 'List'. These values must be separated with a vertical bar.
Nthfile Sequence	1	The specified simulation run time to write to file and to calculate genetic distance matrices. This value is used if nthfile_choice = 'Sequence'. This is the 'by' value in the sequence. For example 1 would create values starting at 0, ending at looptime-1, by every 1 generation. A value

		of 5 would output every 5th generation, etc.
Adult Mortality Percent	100	Percent mortality in the adult population. Use 100 here to specify non-overlapping generations.

3.3 Mating parameters

The following lists the parameters used for the movement of individuals due to mating.

File Header	Example	Description
Mate Movement Number	1	Movement function answer for mating. 1 = Linear, 2 = Inverse Square, 3 = Nearest Neighbor, 4 = Random Mixing, and 5 = Negative Exponential.
Mate Movement parA	1	This is only used for negative exponential y = a*10^-bx and is the parameter a.
Mate Movement parB	1	This is only used for negative exponential y = a*10^-bx and is the parameter b.
Mate Movement Threshold	5	A threshold option (in cost distance units) for how far an individual can search for a mate. You can specify 'max' to consider all individuals for mating movement. You can also place an integer value in front of 'max' to consider a percent cost distance movement for mating. For example '10max' would consider all mating individuals that are within 10 percent of the maximum cost distance on the surface. You can also just specify a specific cost distance value.
Female Replacement	N	If you want females to mate with replacement, then specify 'Y'. If you want females to mate without replacement, then specify 'N'.
Male Replacement	Y	If you want males to mate with replacement, then specify 'Y'. If you want males to mate without replacement, then specify 'N'.
Selfing Answer	N	If you want to allowing selfing (i.e., individuals mate with themselves), then specify 'Y'. If you do not want to allow for selfing, then specify 'N'.
Repro-duction Answer	Y	'Y' for sexual reproduction and 'N' for asexual reproduction. With asexual reproduction, all *n* individuals mate and bear offspring, with mates selected according to the movement function choice and without regard to any gender or mating type. It is important to note that this "asexual" reproduction is functionally the sexual paring of hermaphroditic individuals; thus it is asexual in the sense that there are no distinct sexes, but is sexual in the sense

		that individuals mate with other individuals, exchanging genes in Mendelian reproduction. In sexual reproduction, mated pairs consider male and females with or without replacement.
Repro-duction Age	0	The age at which individuals can start to reproduce. Use with overlapping generations, i.e, oldmortperc not set to 100. This value must be 0 if you are using nonoverlapping generations.

3.4 Dispersal parameters

Here lists the parameters used for the movement of individuals with regards to offspring dispersal.

File Header	Example	Description
Female Dispersal Movement Number	2	This is the function answer for movement for female dispersal. 1 = Linear, 2 = Inverse Square, 3 = Nearest Neighbor, 4 = Random Mixing, and 5 = Negative Exponetial.
Female Dispersal Movement-parA	1	Used only for negative exponential y = a*10^-bx and is the parameter a.
Female Dispersal Movement-parB	1	Used only for negative exponential y = a*10^-bx and is the parameter b.
Female Dispersal Movement Threshold	5	A threshold option (in cost distance units) for how far an individual female offspring can disperse. You can specify 'max' to consider all individuals for mating movement. You can also place an integer value in front of 'max' to consider a percent cost distance movement for mating. For example '10max' would consider all mating individuals that are within 10 percent of the maximum cost distance on the surface. You can also just specify a specific cost distance value.
Male Dispersal Movemen Number	1	This is the function answer for movement for male dispersal. 1 = Linear, 2 = Inverse Square, 3 = Nearest Neighbor, 4 = Random Mixing, and 5 = Negative Exponential.
Male Dispersal Movement-parA	1	Used only for negative exponential y = a*10^-bx and is the parameter a.
Male Dispersal Movement-parB	1	Used only for negative exponential y = a*10^-bx and is the parameter b.
Male Dispersal	10	A threshold option (in cost distance units) for how far an individual male offspring can

Movement-Threshold		disperse. You can specify 'max' to consider all individuals for mating movement. You can also place an integer value in front of 'max' to consider a percent cost distance movement for mating. For example '10max' would consider all mating individuals that are within 10 percent of the maximum cost distance on the surface. You can also just specify a specific cost distance value.

3.5 Offspring parameters

The following lists the parameters that control offspring births and deaths.

File Header	Example	Description
Offspring Number	2	This is the number of offspring each mate pair can have. Choose 1 for a random draw, 2 for Poisson draw, and 3 for a constant number of offspring for each mother or mate pair.
Lambda	5	The parameter value used with Offspring Number. If offno = 1, then lmbda is the max range value between 0 – lmbda to draw randomly from. If offno = 2, then lmbda is the Poisson mean for the litter size. If offno = 3, then lmbda is the constant litter size value.
Female-Percent	50	Percent number of female born in each litter.
Equal Sex Ratio	Y	The answer to have every generation start with equal sex ratios. Possible options include 'N' and 'Y'.
Offspring Mortality Percent	0	Percent mortality in the offspring population.

3.6 Genetic parameters

The following lists the parameters associated with the initialization of the genotypes, mutation rates, and mtDNA option.

File Header	Example	Description
Gene Swap Generation	0	The generation time that genetic information starts to be exchanged.
Mutation Rate	0.0005	The *k*-allele model mutation rate.
Loci	10	The number of loci.
Intialize Genes Answer	random	The choice for how to initialize the genotype for each *n-(x,y)* individuals. If 'random' is entered, then the genotypes get a random assignment and the population is at a maximum genetic diversity. If 'file' is entered, then the genetics get drawn from the allele frequency distribution file

		(specified in next column, allefreqfilename). If 'known' is entered, then the genotypes are directly read from a given known file. This file is very similar to the initial xyfilename, but email Erin Landguth (erin.landguth@mso.umt.edu) for an example file to use for 'Known'.
Allefreq-filename	N	The allele frequency distribution for each locus, used to initialize the model's *n* individual's genotype. If you want to use a frequency distribution file, you must set Initialize Genes Answer to equal 'file' and then enter in the filename in this field. See allelefrequency.csv example file for formatting this file. It is basically a row of allele frequencies and make sure the length of the row equals your starting loci * starting alleles.
Alleles	10	The number of starting alleles per locus.
mtDNA	N	If 'Y', then last locus becomes mtDNA and every offspring inherits this locus from its mother only. If 'N', then regular Mendal inheritance occurs for this last locus.

3.7 CDEVOLVE

The following lists the parameters and surfaces used to simulate natural selection.

File Header	Example	Description
CDEVOLVE Answer	0	This is the answer for how many loci are under selection. Use '0' to turn off CDEVOLVE. Use '1' for natural selection with 1 locus. Use '2' for selection with 2 loci. Alleles must be 2 if 'Y' is entered.
The following are the fitness surfaces for when CDEVOLVE Answer is 1. This corresponds to 1 locus that is under selection. The x,y location of an offspring is matched up with the closest x,y fitness value for the offspring's corresponding fitness surface defined by the genotype that the offspring has. That value then becomes the individual offspring mortality percentage. An offspring becomes more or less fit relative to the other offspring at that generation as a function of its genotype and where it occurs on a surface. These are ASCII formatted files with 6 lines of header information and values that are space delimited and represent percent mortality [0 - 100]. See fitvals.txt for an example fitness surface and its format.		
Fitness AA	Fitvals0	When CDEVOLVE Answer is 1, then this is the offspring viability selection surface for AA. If offspring has AA, then this mortality fitness surface is used.
Fitness Aa	Fitvals0	When CDEVOLVE Answer is 1, then this is the offspring viability selection surface for Aa. If offspring has Aa, then this mortality

		fitness surface is used.
Fitness aa	Fitvals0	When CDEVOLVE Answer is 1, then this is the offspring viability selection surface for aa. If offspring has aa, then this mortality fitness surface is used.
The following are the fitness surfaces for when CDEVOLVE Answer is 2. This corresponds to 2 loci that are under selection. The x,y location of an offspring is matched up with the closest x,y fitness value for the offspring's corresponding fitness surface defined by the genotype that the offspring has. That value then becomes the individual offspring mortality percentage. An offspring becomes more or less fit relative to the other offspring at that generation as a function of its genotype and where it occurs on a surface. These are ASCII formatted files with 6 lines of header information and values that are space delimited and represent percent mortality [0 - 100]. See fitvals.txt for an example fitness surface and its format.		
Fitness AABB	Fitvals0	When CDEVOLVE Answer is 2, then this is the offspring viability selection surface for AABB. If offspring has AABB, then this mortality fitness surface is used.
Fitness AaBB	Fitvals0	When CDEVOLVE Answer is 2, then this is the offspring viability selection surface for AaBB. If offspring has AaBB, then this mortality fitness surface is used.
Fitness aaBB	Fitvals0	When CDEVOLVE Answer is 2, then this is the offspring viability selection surface for aaBB. If offspring has aaBB, then this mortality fitness surface is used.
Fitness AABb	Fitvals0	When CDEVOLVE Answer is 2, then this is the offspring viability selection surface for AABb. If offspring has AABb, then this mortality fitness surface is used.
Fitness AaBb	Fitvals0	When CDEVOLVE Answer is 2, then this is the offspring viability selection surface for AaBb. If offspring has AaBb, then this mortality fitness surface is used.
Fitness aaBb	Fitvals0	When CDEVOLVE Answer is 2, then this is the offspring viability selection surface for aaBb. If offspring has aaBb, then this mortality fitness surface is used.
Fitness AAbb	Fitvals0	When CDEVOLVE Answer is 2, then this is the offspring viability selection surface for AAbb. If offspring has AAbb, then this mortality fitness surface is used.
Fitness Aabb	Fitvals0	When CDEVOLVE Answer is 2, then this is the offspring viability selection surface for Aabb. If offspring has Aabb, then this mortality fitness surface is used.

Fitness aabb	Fitvals0	When CDEVOLVE Answer is 2, then this is the offspring viability selection surface for aabb. If offspring has aabb, then this mortality fitness surface is used.

3.8 CDINFECT

These parameters are used with the module CDINFECT. Currently only vertical transmission is assumed. Future development will include horizontal transmission parameters. This a beta test module.

File Header	Example	Description
CDINFECT	N	This is the infection parameter answer. This tracks vertical transmission in the population. If 'Y', then a random status infection (0 or 1) is created and initialized for each individual. If 'N', then the status 0 is created for all individuals and initialized. A column in grid.csv denotes the infection status at each generation for every individual.
Trans-mission Probability	0.5	This is the transmission probability for if a parent has the infection the chance that the infection will be passed along to the offspring.

3.9 CDCLIMATE

These are the parameters that control the dynamic landscape functionality within CDPOP. A generation time is specified and input cost distance matrices are then read into the program and used in simulations.

File Header	Example	Description
CDCLIMATE	N	This is the dynamic landscape answer. If 'Y', then a new cost distance matrix will be read in at a specified generation time in the next column.
CDCLIMATE Generation	5|10	The generation time that the next cost distance matrix will be read in at. You can specify multiple generations by separating each generation to read in the next cost distance matrix by '|'.
Future Mating CDMatrix	ED-cdmatrix16|ED-Cdmatrix16	A *[nxn]* future cost distance matrix for mating movement, where *n* is the number of grid values (or individuals) on the landscape. This is a comma delimited file format. Note that this file must be the same size as the initial cost distance matrices used in the simulations. If you are using multiple surfaces (i.e., you specified multiple cdclimate generations by separating them with '|'), then you will also need to specify multiple cost distance matrices here as well by using a '|'.

Future Dispersal CDMatrix	ED-cdmatrix16| ED-Cdmatrix16	A *[nxn]* future cost distance matrix for dispersal movement, where *n* is the number of grid values (or individuals) on the landscape. This is a comma delimited file format. Note that this file must be the same size as the initial cost distance matrices used in the simulations. If you are using multiple surfaces (i.e., you specified multiple cdclimate generations by separating them with '|'), then you will also need to specify multiple cost distance matrices here as well by using a '|'.

4 Output Files

4.1 Individual-based output files

The following is a list of output options from CDPOP, including options to calculate cost distance matrices, a Euclidean distance matrix, genetic distance matrices, and genotype formatting.

File Header	Example	Description
EDmatans	N	This is the Euclidean distance matrix answer. If 'Y', then the Euclidean distance matrix used for the *n* original grid locations is calculated. If 'N' is entered, then this matrix is not created. This matrix will be called EDmatrix.csv and outputted in the respective batch and Monte Carlo folder.
GDmatans	Dps	This is the genetic distance matrix answer. The genetic distance matrix used for the *n* original grid locations for specified generation time of the simulation run is calculated. Enter 'braycurtis' for the Bray-Curtis distance measure, 'Dps' for the proportion of shared alleles, or 'Da' for Nei's genetic distance. Specify 'N' here if you do not want to calculate these matrices. Note that Nei's genetic distance takes the longest to calculate and may decrease your total CDPOP simulation time. All are slightly expensive in computation time and we are working on speeding these up.
Gridformat	General	This is the genotype output format option. The format for the genotype output is specified by entering 'general' for a general genotype output or 'cdpop' for the cdpop genotype output. The general format will follow Locus1a, Locus1b, Locus2a, Locus2b, …, LocusNa, LocusNb. The cdpop format lists the genotypes with values for each allele -> either 0, 1, or 2.

4.2 Population-based output files

In addition, a output.csv is automatically created for each batch and Monte Carlo run. These are population based metrics calculated at each generation. The following is a summary of each calculation:

- Year – This is the generation time.
- Population – The total population in each generation. This is always constant right now. Demographics and population dynamics is coming soon…
- ToTFemales – The total number of females in each generation.
- ToTMales – The total number of males in each generation.
- BreedFemales – The total number of breeding age females in each generation.
- BreeMales – The total number of breeding age females in each generation.
- Emigrants – The number of left over offspring at each generation. These get deleted from the program and are no longer considered.
- Immigrants – The number of immigrants that came in to fill up a habitat location or a grid location.
- Births – The number of offspring born at that generation.
- Deaths – The number of deaths of the adult population, not the offspring.
- Alleles – This is the total number of unique alleles at each generation. This value can be calculated automatically for specified subpopulations if different subpopulations were designated in the initial xyfilename. If there are '|', then the first value corresponds to the total alleles in the population and subpopulation values follow after.
- He - This is the expected heterozygosity value at each generation. This value can be calculated automatically for specified subpopulations if different subpopulations were designated in the initial xyfilename. If there are '|', then the first value corresponds to the total He in the population and subpopulation values follow after.
- Ho - This is the observed heterozygosity value at each generation. This value can be calculated automatically for specified subpopulations if different subpopulations were designated in the initial xyfilename. If there are '|', then the first value corresponds to the total Ho in the population and subpopulation values follow after.
- Mutations – The total number of mutations at each generation.
- MateDistED – The average Euclidean distance individuals travel to mate.
- DispDistED – The average Euclidean distance individual offspring disperse from their natal location.
- MateDistCD – The average cost distance individuals travel to mate.
- DispDistCD – The average cost distance individual offspring disperse from their natal location.
- P1 – This is the allele frequency of A (used for CDEVOLVE).

- P2 – This is the allele frequency of a (used for CDEVOLVE).
- q1 – This is the allele frequency of B (used for CDEVOLVE).
- q2 – This is the allele frequency of b (used for CDEVOLVE).
- Infected – The number of individuals that are infected at each generation (used for CDINFECT).

5 General issues

5.1 How to obtain CDPOP

The program is freeware and can be downloaded at http://cel.dbs.umt.edu/software/CDPOP/ with information for users, including manual instructions, FAQ, publications, ongoing research, and developer involvement.

5.2 Debugging and troubleshooting

For help with installation problems please check first for postings at our web site. Otherwise, please report problems including any bugs, to me at erin.landguth@mso.umt.edu.

5.3 How to cite CDPOP

This program was developed by Erin Landguth with help from Brian Hand, Joe Glassy, Sam Cushman, and Tyler Julian. GUI development was done by Mike Jacobi. The reference to cite is as follows, substituting the version number:
Landguth EL, Cushman SA, Jacobi M (2010) CDPOP: A spatially-explict cost distance populatin genetics program. Molecular Ecology Resources. 10:156-161 "Version X".

5.4 Disclaimer

The software is in the public domain, and the recipient may not assert any proprietary rights thereto nor represent it to anyone as other than a University of Montana-produced program (version 1.x). CDPOP is provided "as is" without warranty of any kind, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The user assumes all responsibility for the accuracy and suitability of this program for a specific application. In no event will the authors or the University be liable for any damages, including lost profits, lost savings, or other incidental or consequential damages arising from the use of or the inability to use this program.

We strongly urge you to read the entire documentation before ever running CDPOP. We wish to remind users that we are not in the commercial software marketing business. We are scientists who recognized the need for a tool like CDPOP to assist us in our research on landscape ecology issues. Therefore, we do not wish to spend a great deal of time consulting on trivial matters concerning the use of CDPOP. However, we do recognize an obligation to provide some level of information support. Of course, we welcome and encourage your criticisms and suggestions about the program at all times. We will welcome questions about how to run CDPOP or interpret the output only after you have read the

entire documentation. This is only fair and will eliminate many trivial questions. Finally, we are always interested in learning about how others have applied CDPOP in ecological investigation and management application. Therefore, we encourage you to contact us and describe your application after using CDPOP.

We hope that CDPOP is of great assistance in your work and we look forward to hearing about your applications.

6 References

Allendorf,F.W. and Luikart,G. (2007) Conservation and the genetics of populations. Blackwell, Malden, MA.

Bowcock,A.M. *et al.* (1994) High resolution of human evolutionary trees with polymorphic micorsatellites. *Nature*. **368**, 455-457.

Cushman,S.A. *et al.* (2006) Gene Flow in Complex Landscapes: Testing Multiple Hypotheses with Casual Modeling. *The American Naturalist* **168**, 486-499.

Cushman,S.A. and Landguth,E.L. (2010) Spurious correlations and inferences in landscape genetics. *Molecular Ecology*, **19**, 3592-3602.

Holderegger,R. and Wagner,H.H. (2006) A brief guide to Landscape Genetics. *Landscape Ecology* **21**, 793-796.

Landguth,E.L. and Cushman,S.A. (2010) CDPOP: A spatially-explicit cost distance population genetics program, *Molecular Ecology Resources*, **10**, 156-161.

Landguth,E.L. *et al.* (2010a) Quantifying the lag time to detect barriers in landscape genetics. *Molecular Ecology*, **19**, 4179-4191.

Landguth,E.L. *et al.* (2010b) Relationships between migration rates and landscape resistance assessed using individual-based simulations. *Molecular Ecology Resources*, **10**, 854-862.

Legendre,P. and Legendre,L. (1998) Numerical ecology. 2nd English ed. Elsevier,Amsterdam.

McRae,B.H. and Beier,P. (2007) Circuit theory predicts gene flow in plant and animal populations. *Proceedings of the National Academy of Science USA* **104**, 19885-19890.

Nei,M. *et al.* (1983) Accuracy of estimated phylogenetic trees from molecular data. *Journal of Molecular Evolution* **19**,153–170.

Ray,N. (2005) PATHMATRIX: a GIS tool to compute effective distances among samples. *Molecular Ecology Notes* **5**, 177-180.

Storfer,A. *et al.* (2010) Landscape genetics: where are we now? *Molecular Ecology*, **19**,3496–3514.

Wright,S. (1932) The roles of mutation, inbreeding, crossbreeding, and selection in evolution, *Proceedings XI International Congress of Genetics*, **1**, 356-366.

7 Acknowledgements

This research was supported in part by funds provided by the Rocky Mountain Research Station, Forest Service, U.S. Department of Agriculture and by the National Science Foundation grant #DGE-0504628.